Journal of Statistical Software

February 2024, Volume 108, Issue 5. doi: 10.18637/jss.v108.i05

melt: Multiple Empirical Likelihood Tests in R

Eunseop Kim The Ohio State University

Steven N. MacEachern The Ohio State University

Mario Peruggia The Ohio State University

Abstract

Empirical likelihood enables a nonparametric, likelihood-driven style of inference without relying on assumptions frequently made in parametric models. Empirical likelihood-based tests are asymptotically pivotal and thus avoid explicit studentization. This paper presents the R package **melt** that provides a unified framework for data analysis with empirical likelihood methods. A collection of functions are available to perform multiple empirical likelihood tests for linear and generalized linear models in R. The package **melt** offers an easy-to-use interface and flexibility in specifying hypotheses and calibration methods, extending the framework to simultaneous inferences. Hypothesis testing uses a projected gradient algorithm to solve constrained empirical likelihood optimization problems. The core computational routines are implemented in C++, with **OpenMP** for parallel computation.

Keywords: empirical likelihood, generalized linear models, hypothesis testing, optimization, R.

1. Introduction

The likelihood is an essential component of statistical inference. In a nonparametric or semi-parametric setting, where the quantity of interest is finite-dimensional, the maximum likelihood approach is not applicable since the underlying data-generating distribution is left unspecified. A popular approach in this context is the method of moments or the two-step generalized method of moments (GMM, Hansen 1982) where only partial information is specified by moment conditions. Various one-step alternatives to GMM have been proposed over the last decades in the statistics and econometrics literature (see, e.g., Efron 1981; Imbens 1997; Newey and Smith 2004).

One such alternative is empirical likelihood (EL, Owen 1988, 1990; Qin and Lawless 1994). EL defines a likelihood function by profiling a nonparametric likelihood subject to the moment restrictions. While it is nonparametric in nature, some desirable properties of parametric

likelihood apply to EL. Most notably, the EL ratio functions have limiting chi-square distributions under certain conditions. Without explicit studentization, confidence regions for the parameters can be constructed in much the same way as with a parametric likelihood. As the name suggests, however, the empirical distribution of the data determines the shape of the confidence region. Also, coverage accuracy of the confidence region can further be improved in principle, since EL is Bartlett-correctable (DiCiccio, Hall, and Romano 1991). In terms of estimation, the standard expansion argument (e.g., Yuan and Jennrich 1998; Jacod and Sørensen 2018) establishes the consistency and asymptotic normality of the maximum empirical likelihood estimator (MELE). Moreover, Newey and Smith (2004) showed that the MELE generally has a smaller bias than its competitors and achieves higher-order efficiency after bias correction. EL methods have been extended to other areas, including linear models (Owen 1991), generalized linear models (Kolaczyk 1994; Chen and Cui 2003), survival analysis (Li, Li, and Zhou 2005), time series models (Kitamura 1997; Nordman and Lahiri 2014), and high-dimensional data analysis (Chen, Peng, and Qin 2009; Hjort, McKeague, and Van Keilegom 2009). For an overview of EL and its applications, see Owen (2001) and Chen and Van Keilegom (2009).

In the R language (R Core Team 2023), some software packages implementing EL and related methods are available from the Comprehensive R Archive Network (CRAN). The **emplik** package (Zhou 2023) provides a wide range of functions for analyzing censored and truncated data with EL. Confidence intervals for a one-dimensional parameter can also be constructed. Other examples and applications of the package can be found in Zhou (2015). The **emplik2** package (Barton 2022) is an extension for the two sample problems. Both packages cover the methods for the mean with uncensored data, which is the simplest case in terms of computation. In addition, the **EL** package (Valeinis and Cers 2022) performs EL tests for the difference between two sample means and the difference between two smoothed Huber estimators. The **elhmc** package (Kien, Chaudhuri, and Wei 2017) contains a single function `ELHMC()` for Hamiltonian Monte Carlo sampling in Bayesian EL computation (Chaudhuri, Mondal, and Yin 2017). The **ELCIC** package (Shen and Wang 2023) develops an EL-based consistent information criterion in a model selection framework. In a broader context of GMM and generalized empirical likelihood (Smith 1997), a few packages can be used for estimation with EL. The **gmm** package (Chaussé 2010) provides flexibility in specifying moment conditions. Other than GMM and EL, continuous updating (Hansen, Heaton, and Yaron 1996) and several estimation methods that belong to the family of generalized empirical likelihood are available. The **gmm** package has been superseded by the **momentfit** package (Chaussé 2023), which adds exponential tilting (Kitamura and Stutzer 1997) estimation and methods for constructing two-dimensional confidence regions.

This paper presents the R package **melt** (Kim 2024) that performs multiple empirical likelihood tests for regression analysis. The primary focus of the package is on linear and generalized linear models, perhaps most commonly used with the `lm()` and `glm()` functions in R. The package considers only just-identified models where the number of moment conditions equals the number of parameters. Typical linear models specified by `formula` objects in R are just-identified. In this case, the MELE is identical to the maximum likelihood estimator, and the estimate is easily obtained using `lm.fit()` or `glm.fit()` in the **stats** package. The fitted model then serves as a basis for testing hypotheses, which is a core component of the package. EL-based tests do not involve standard errors and `vcov()` methods since they are asymptotically pivotal and thus avoid explicit studentization. For this reason it is

challenging to incorporate EL methods directly into other packages that perform inferences for parametric models using `vcov()`. We aim to bridge the gap and provide an easy-to-use interface that enables applying the methods to tasks routinely accomplished in R. Standard tests performed by `summary.lm()` and `summary.glm()` methods, such as significance tests of the coefficients and an overall F test or a chi-square test, are available. Furthermore, in line with `lht()` in the **car** package (Fox and Weisberg 2019) or `glht()` in the **multcomp** package (Hothorn, Bretz, and Westfall 2008), the user can specify linear hypotheses to be tested. Multiple testing procedures are provided to control the family-wise error rate. Constructing confidence intervals and detecting outliers in a fitted model can also be done, adding more options for data analysis. Note that all the tests and methods rely on EL and its asymptotic properties. Although conceptually advantageous over parametric methods, this can lead to poor finite sample performance. Therefore, several calibration techniques are implemented in **melt** to mitigate this drawback of EL.

The rest of the paper is organized as follows. Section 2 describes EL methods and computational aspects of testing hypotheses with EL. Section 3 provides an overview of the **melt** package. Section 4 shows the basic usage of **melt** with implementation details. Section 5 presents an application to pest control experiments. We conclude with a summary and directions for future development in Section 6.

2. Background

2.1. Empirical likelihood framework

We describe a general framework for EL formulation. Suppose we observe independent and identically distributed (i.i.d.) p-dimensional random variables $X_1, \ldots, X_n$ from a distribution P. Consider a parameter $\theta \equiv \theta(P) \in \Theta$ and an estimating function $g(X_i, \theta)$ that takes its values in $\mathbb{R}^p$ and satisfies the following moment condition:

$$\mathsf{E}[g(X_i, \theta)] = 0, \tag{1}$$

where the expectation is taken with respect to P. Without further information on P, we restrict our attention to a family of multinomial distributions supported on the data. The nonparametric likelihood is given by

$$L(P) = \prod_{i=1}^{n} P(\{X_i\}) = \prod_{i=1}^{n} p_i,$$

and the empirical distribution P_n maximizes L with $L(P_n) = n^{-n}$. Then the (profile) EL ratio function is defined as

$$R(\theta) = \max_{p_i} \left\{ \prod_{i=1}^{n} np_i : \sum_{i=1}^{n} p_i g(X_i, \theta) = 0,\ p_i \geq 0,\ \sum_{i=1}^{n} p_i = 1 \right\}, \tag{2}$$

with $L(\theta) = \prod_{i=1}^{n} p_i$ denoting the corresponding EL function. Ties in the data do not affect the EL formulation both computationally and theoretically. We can assign probability weight p_i to each observation as if there are no ties and still obtain the same EL ratio value (Owen 2001, Section 2.3). The profiling removes all the nuisance parameters, the p_is attached to

the data, yielding a p-dimensional subfamily indexed by θ. Note that the data determine the multinomial distributions; thus, the reduction to a subfamily does not correspond to a parametric model. See DiCiccio and Romano (1990) for a detailed discussion of how the reduction connects to the notion of least favorable families (Stein 1956).

We maximize $\prod_{i=1}^n np_i$, or equivalently $\sum_{i=1}^n \log(np_i)$, subject to the constraints in Equation 2. Implicit within these constraints is a condition known as the "convex hull constraint", stipulating that the convex hull of the points $\{g(X_i,\theta)\}_{i=1}^n$ must contain the zero vector. Failure to meet this constraint renders the problem infeasible, forcing some p_is to be negative to satisfy the condition $\sum_{i=1}^n p_i g(X_i,\theta) = 0$. When the constraint is met, the problem admits a unique interior solution since the objective function is concave and the feasible set is convex. Using the method of Lagrange multipliers, we write

$$\mathcal{L}(p_1,\ldots,p_n,\lambda,\gamma) = \sum_{i=1}^n \log(np_i) - n\lambda^\top \sum_{i=1}^n p_i g(X_i,\theta) + \gamma\left(\sum_{i=1}^n p_i - 1\right),$$

where λ and γ are the multipliers. Differentiating $\mathcal{L}$ with respect to its arguments and setting the derivatives to zero gives $\gamma = -n$. Then the solution is given by

$$p_i = \frac{1}{n}\frac{1}{1+\lambda^\top g(X_i,\theta)}, \tag{3}$$

where $\lambda \equiv \lambda(\theta)$ satisfies

$$\frac{1}{n}\sum_{i=1}^n \frac{g(X_i,\theta)}{1+\lambda^\top g(X_i,\theta)} = 0. \tag{4}$$

Instead of solving the nonlinear Equation 4, we solve the dual problem with respect to λ. Substituting the expression for p_i in Equation 3 into $\sum_{i=1}^n \log(np_i)$ gives

$$\log\left(R(\theta)\right) = -\sum_{i=1}^n \log\left(1+\lambda^\top g(X_i,\theta)\right) =: r(\lambda). \tag{5}$$

Now consider minimizing $r(\lambda)$ subject to $1+\lambda^\top g(X_i,\theta) \geq 1/n$ for $i = 1,\ldots,n$ with θ fixed. This is a convex optimization problem, where the constraints correspond to the condition that $0 \leq p_i \leq 1$ for all i. Next, we remove the constraints by employing a pseudo logarithm function (Owen 1990)

$$\log_\star(x) = \begin{cases} \log(x), & \text{if } x \geq 1/n, \\ -n^2x^2/2 + 2nx - \log(n) - 3/2, & \text{if } x < 1/n. \end{cases} \tag{6}$$

Minimizing $r_\star(\lambda) = -\sum_{i=1}^n \log_\star(1+\lambda^\top g(X_i,\theta))$ without the constraints does not affect the solution and the Newton-Raphson method can be applied to find it. If the convex hull constraint is violated, the algorithm does not converge with $\|\lambda\|$ increasing as the iteration proceeds. Hence, it can be computationally more efficient to minimize $r_\star(\lambda)$ first to get $\log(R(\theta))$ and indirectly check the convex hull constraint by observing λ and p_is. Note that EL is maximized when $\lambda = 0$ and $p_i = 1/n$ for all i. It follows from Equation 4 that $\hat{\theta}$, the MELE, is obtained by solving the estimating equations $\sum_{i=1}^n g(X_i,\theta) = 0$. The existence, uniqueness, and asymptotic properties of $\hat{\theta}$ are well established in the literature.

Assume that there exists a true parameter value $\theta_0 \in \Theta$ that is the unique solution to the moment condition in Equation 1. Similar to the parametric likelihood method, define minus

twice the empirical log-likelihood ratio function as $l(\theta) = -2\log(R(\theta))$. Under regularity conditions, it is known that a nonparametric version of Wilks' theorem holds. That is, $l(\theta_0) \to_d \chi^2_p$ as $n \to \infty$, where χ^2_p is the chi-square distribution with p degrees of freedom. See, e.g., Qin and Lawless (1994) for proof and the treatment of more general cases, including the over-identified ones. For a level $\alpha \in (0,1)$, let $\chi^2_{p,\alpha}$ be the $100(1-\alpha)\%$th percentile of χ^2_p. Since $P(l(\theta_0) \le \chi^2_{p,\alpha}) \to 1-\alpha$, an asymptotic $100(1-\alpha)\%$ confidence region for θ can be obtained as

$$\left\{\theta : l(\theta) \le \chi^2_{p,\alpha}\right\}. \tag{7}$$

Often the chi-square calibration is unsatisfactory due to slow convergence, especially when n is small. We review other calibration methods that address this issue. First, consider EL for the mean with $g(X_i,\theta) = X_i - \theta$ and $\theta_0 = \mathsf{E}[X_i]$. The MELE is the sample average $\bar{X}$ since $\sum_{i=1}^n (X_i - \bar{X}) = 0$. It can be shown that $l(\theta_0) = n(\bar{X}-\theta_0)^\top V^{-1}(\bar{X}-\theta_0) + o_P(1)$, where $V = n^{-1}\sum_{i=1}^n (X_i-\theta_0)(X_i-\theta_0)^\top$. Let $S = (n-1)^{-1}\sum_{i=1}^n (X_i-\bar{X})(X_i-\bar{X})^\top$ and define a Hotelling's T squared statistic as $T^2 = n(\bar{X}-\theta_0)^\top S^{-1}(\bar{X}-\theta_0)$. It follows that

$$n(\bar{X}-\theta_0)^\top V^{-1}(\bar{X}-\theta_0) = nT^2/(T^2+n-1) = T^2 + O_P(n^{-1}).$$

This suggests that we can use $p(n-1)F_{p,n-p,\alpha}/(n-p)$ in place of $\chi^2_{p,\alpha}$, where $F_{p,n-p}$ is a F distribution with p and $n-p$ degrees of freedom. The F calibration yields a larger critical value than the chi-square calibration, which leads to a better coverage probability of the confidence region in Equation 7. Next, a more generally applicable calibration method is the Bartlett correction. Based on the Edgeworth expansion, it requires Cramér's condition and the existence of finite higher moments of $g(X_i,\theta)$. The correction is given by a scale multiple of $\chi^2_{p,\alpha}$ as $(1+a/n)\chi^2_{p,\alpha}$ with an unknown constant a. In practice, the Bartlett correction involves getting a consistent estimate $\hat{a}$ with plug-in sample moments. The coverage error of the Bartlett corrected confidence region is reduced from $O(n^{-1})$ to $O(n^{-2})$ (DiCiccio *et al.* 1991), which is unattainable by the F calibration. Another effective calibration method is the bootstrap. Let $\widetilde{\mathcal{X}}_n = \{\widetilde{X}_1, \ldots, \widetilde{X}_n\}$ denote the null-transformed data such that $\mathsf{E}_{P_n}[g(\widetilde{X}_i,\theta)] = n^{-1}\sum_{i=1}^n g(\widetilde{X}_i,\theta) = 0$, so Equation 1 holds for $\widetilde{\mathcal{X}}_n$ with P_n. Define a bootstrap sample $\widetilde{\mathcal{X}}^*_n = \{\widetilde{X}^*_1, \ldots, \widetilde{X}^*_n\}$ as i.i.d. observations from $\widetilde{\mathcal{X}}_n$. We can compute the bootstrap EL ratio $l^*(\theta)$ with $\widetilde{\mathcal{X}}^*_n$ in the same way as before. The critical value, the $100(1-\alpha)\%$th percentile of the distribution of $l^*(\theta)$, can be approximated using a large number, say B, of bootstrap samples. As an example, we may set $\widetilde{X}_i = X_i - \bar{X} + \theta$ when $g(X_i,\theta) = X_i - \theta$. This is equivalent to computing $l^*(\bar{X})$ with a bootstrap sample from the observed data directly. The $O(n^{-2})$ coverage error can also be achieved by the bootstrap calibration (Hall and Scala 1990).

Although EL does not require full model specification, it is not entirely free of misspecification concerns. Developing diagnostic measures for EL is still an open problem, and we briefly introduce the technique of empirical likelihood displacement (ELD, Lazar 2005). Much like the concept of likelihood displacement (Cook 1986), ELD can be used to detect influential observations or outliers. With the MELE $\hat{\theta}$ from the complete data, consider reduced data with the ith observation deleted and the corresponding MELE estimate $\hat{\theta}_{(i)}$, Then ELD is defined as

$$\mathrm{ELD}_i = 2\left(L(\hat{\theta}) - L(\hat{\theta}_{(i)})\right), \tag{8}$$

where $\hat{\theta}_{(i)}$ is plugged into the original EL function $L(\theta)$. If ELD_i is large, the ith observation is an influential point and can be inspected as a possible outlier. See Zhu, Ibrahim, Tang, and Zhang (2008) for other diagnostic measures for EL.

2.2. Empirical likelihood for linear models

We now turn our attention to linear models, which are the main focus of **melt**. Suppose we have independent observations $\{(Y_i, X_i)\}_{i=1}^n$, where Y_i is the univariate response and X_i is the p-dimensional vector of covariates (including the intercept, if any). For illustrative purposes, we consider X_i fixed and do not explicitly distinguish between random and fixed designs. See Kitamura, Tripathi, and Ahn (2004) for formal methods for models with conditional moment restrictions. For standard linear regression models, assume that

$$\mathsf{E}[Y_i] = \mu_i, \quad \mathsf{VAR}[Y_i] = \sigma_i^2, \quad i = 1, \ldots, n,$$

where $\mu_i = X_i^\top \theta_0$ for some true value $\theta_0 \in \mathbb{R}^p$. Since θ_0 minimizes $\mathsf{E}[(Y_i - X_i^\top \theta)^2]$, we have the following moment conditions

$$\mathsf{E}[(Y_i - X_i^\top \theta) X_i] = 0, \quad i = 1, \ldots, n,$$

and the estimating equations

$$\sum_{i=1}^n (Y_i - X_i^\top \theta) X_i = 0.$$

Let $Z_i = (Y_i, X_i)$ and $g(Z_i, \theta) = (Y_i - X_i^\top \theta) X_i$. The $g(Z_i, \theta)$s are independent with the mean $(\mu_i - X_i^\top \theta) X_i$ and variance $\sigma_i^2 X_i X_i^\top$. Following the steps in Section 2.1, we can compute the EL ratio function

$$R(\theta) = \max_{p_i} \left\{ \prod_{i=1}^n n p_i : \sum_{i=1}^n p_i g(Z_i, \theta) = 0, \ p_i \geq 0, \ \sum_{i=1}^n p_i = 1 \right\}. \tag{9}$$

Under mild moment conditions it follows that $l(\theta_0) \to_d \chi_p^2$. Note also from Equation 9 that the least square estimator $\hat{\theta}$ is the MELE of θ, with $L(\hat{\theta}) = n^{-n}$ and $R(\hat{\theta}) = 1$.

Next, generalized linear models assume that

$$\mathsf{E}[Y_i] = \mu_i, \quad G(\mu_i) = X_i^\top \theta, \quad \mathsf{VAR}[Y_i] = \phi V(\mu_i), \quad i = 1, \ldots, n,$$

where G and V are known link and variance functions, respectively, and ϕ is an optional dispersion parameter. EL for generalized linear models builds upon quasi-likelihood methods (Wedderburn 1974). The log quasi-likelihood for Y_i is given by

$$Q(Y_i, \mu_i) = \int_{Y_i}^{\mu_i} \frac{Y_i - t}{\phi V(t)} dt.$$

Differentiating $Q(Y_i, \mu_i)$ with respect to θ yields the quasi-score

$$\frac{H'(X_i^\top \theta) \left(Y_i - H(X_i^\top \theta) \right)}{\phi V \left(H(X_i^\top \theta) \right)} X_i =: g_1(Z_i, \theta),$$

where H denotes the inverse link function. From $\mathsf{E}[g_1(Z_i, \theta_0)] = 0$ for $i = 1, \ldots, n$, we get the estimating equations

$$\sum_{i=1}^n g_1(Z_i, \theta) = 0.$$

Then the EL ratio function can be derived as in Equation 9 with the same asymptotic properties. It can be seen that the MELE of θ is the same as the quasi-maximum likelihood estimator. When overdispersion is present with unknown ϕ, we introduce another estimating function based on the squared residuals. Let $\eta = (\theta, \phi)$ and

$$g_2(Z_i, \eta) = \frac{\left(Y_i - H(X_i^\top \theta)\right)^2}{\phi^2 V\left(H(X_i^\top \theta)\right)} - \frac{1}{\phi},$$

where $\mathsf{E}[g_2(Z_i, \eta_0)] = 0$ for some $\eta_0 = (\theta_0, \phi_0)$. Then the MELE of ϕ is

$$\hat{\phi} = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(Y_i - H(X_i^\top \hat{\theta})\right)^2}{V\left(H(X_i^\top \hat{\theta})\right)}. \tag{10}$$

We compute the EL ratio function with this additional constraint as

$$R(\eta) = \max_{p_i} \left\{ \prod_{i=1}^{n} n p_i : \sum_{i=1}^{n} p_i g_1(Z_i, \eta) = 0,\ \sum_{i=1}^{n} p_i g_2(Z_i, \eta) = 0,\ p_i \geq 0,\ \sum_{i=1}^{n} p_i = 1 \right\}.$$

The computation is straightforward since the number of parameters equals the number of constraints on the estimating functions. Confidence regions for θ can be constructed by applying a calibration method to $l(\theta)$. One advantage of using EL for linear models is that the confidence regions have data-driven shapes and orientations.

2.3. Hypothesis testing with empirical likelihood

As seen in Section 2.2, it is easy to compute the MELE and evaluate the EL ratio function at a given value for linear models. Conducting significance tests, or hypothesis testing in general, is often the main interest when using a linear model. The EL method can be naturally extended to testing hypotheses by imposing appropriate constraints on the parameter space Θ (Qin and Lawless 1995; Adimari and Guolo 2010). Consider a null hypothesis $\mathcal{H}$ corresponding to a nonempty subset of Θ through a smooth q-dimensional function h such that $\mathcal{H} = \{\theta \in \Theta : h(\theta) = 0\}$. With additional conditions on $\mathcal{H}$ and h, it can be shown that

$$\inf_{\theta : h(\theta) = 0} l(\theta) \to_d \chi_q^2 \tag{11}$$

under the null that $\theta_0 \in \mathcal{H}$. In practice, computing the solution in Equation 11 is a nontrivial task. Recall that the convex hull constraint restricts the domain of $l(\theta)$ to $\Theta_n := \{\theta \in \Theta : 0 \in \mathrm{Conv}_n(\theta)\}$, where $\mathrm{Conv}_n(\theta)$ denotes the convex hull of $\{g(Z_i, \theta)\}_{i=1}^n$ with an estimating function g. Except for a few cases, both $l(\theta)$ and Θ_n are non-convex in θ, and fully identifying Θ_n can be even more challenging than the constrained minimization problem itself. Given that the solution can only be obtained numerically by an iterative process, it is essential to monitor the entire solution path in $\Theta_n \cap \mathcal{H}$. Another difficulty is in the nested optimization structure. The Lagrange multiplier λ needs to be updated for each update of θ, which amounts to solving an inner layer of optimization in Equation 5 at every step. It is clear that no single method can be applied to all estimating functions and hypotheses. Tang and Wu (2014) proposed a nested coordinate descent algorithm for general constrained EL problems, where the outer layer is optimized with respect to θ with λ fixed. After some algebra, we obtain for $\theta \in \Theta_n$ the gradient of the EL ratio function

$$\nabla \log\left(R(\theta)\right) = -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + \lambda^\top g(Z_i, \theta)} \partial_\theta g(Z_i, \theta) \lambda, \tag{12}$$

Algorithm 1: Constrained empirical likelihood optimization using the projected gradient descent.

Data: $X_1, \ldots, X_n$.

Input : $\theta_0 \in \Theta_n \cap \mathcal{H}$, $\lambda_0 = \lambda(\theta_0)$, m (outer layer maximum number of iterations), ϵ (outer layer convergence tolerance), m_l (inner layer maximum number of iterations), ϵ_l (inner layer convergence tolerance), and P.

Output: Optimal θ that minimizes $l(\theta)$ subject to the constraint $\mathcal{H}$ and the corresponding λ.

$\theta \leftarrow \theta_0$; `// Assume that the initial value satisfies the constraints`
$\lambda \leftarrow \lambda_0$;
for $i \leftarrow 1$ **to** m **do** `// Outer layer`
  $\theta_{\text{temp}} \leftarrow \theta$;
  $\lambda_{\text{temp}} \leftarrow \lambda$;
  $\gamma \leftarrow 2$;
  while $l(\theta_{\text{temp}}) \geq l(\theta)$ **do**
    $\gamma \leftarrow \gamma/2$;
    $\Delta \leftarrow -\gamma P \nabla l(\theta)$;
    $\theta_{\text{temp}} \leftarrow \theta + \gamma\Delta$;
    $\lambda_{\text{temp}} \leftarrow 0$;
    for $j \leftarrow 1$ **to** m_l **do** `// Inner layer`
      $\Delta_l \leftarrow -\left(\nabla^2 r_\star(\lambda_{\text{temp}})\right)^{-1} \nabla r_\star(\lambda_{\text{temp}})$;
      $\gamma_l \leftarrow 1$;
      while $r_\star(\lambda_{\text{temp}} + \gamma_l\Delta_l) > r_\star(\lambda_{\text{temp}})$ **do**
        $\gamma_l \leftarrow \gamma_l/2$;
      $\delta_l \leftarrow \|\lambda_{\text{temp}}\|$;
      $\lambda_{\text{temp}} \leftarrow \lambda_{\text{temp}} + \gamma_l\Delta_l$;
      if $\|\gamma_l\Delta_l\| < \epsilon_l\delta_l + \epsilon_l^2$ **then**
        break;
      else
        $j \leftarrow j + 1$;
  $\delta \leftarrow \|\theta_{\text{temp}}\|$;
  $\theta \leftarrow \theta_{\text{temp}}$;
  $\lambda \leftarrow \lambda_{\text{temp}}$;
  if $\|P\nabla l(\theta)\| < \epsilon$ **or** $\|\gamma\Delta\| < \epsilon\delta + \epsilon^2$ **then**
    break;
  else
    $i \leftarrow i + 1$;
return θ and λ;

where $\partial_\theta g(Z_i, \theta)$ represents the Jacobian matrix of $g(Z_i, \theta)$. Observe that the expression does not involve any derivatives with respect to λ. In order to reduce the computational complexity, we focus only on linear hypotheses of the form

$$\mathcal{H} = \{\theta \in \Theta : L\theta = r\}, \tag{13}$$

where L is a $q \times p$ matrix and r is a q-dimensional vector. We use the projected gradient descent approach to obtain a local minimum of $l(\theta)$ in Equation 11. The projected gradient of $l(\theta)$ can be computed from Equation 12 with the orthogonal projector matrix $P = I_p - L^\top (LL^\top)^{-1} L$, where I_p denotes the $p \times p$ identity matrix. Then it would take a relatively small number of iterations for convergence, reducing the required number of inner layer updates of λ. The pseudo code is shown in Algorithm 1.

Controlling the type 1 error rate is necessary when testing multiple hypotheses simultaneously. Recently there has been interest in multiplicity-adjusted test procedures for Wald-type test statistics that asymptotically have a multivariate chi-square distribution under the global null hypothesis (Dickhaus and Royen 2015; Dickhaus and Sirotko-Sibirskaya 2019). Kim, MacEachern, and Peruggia (2023) proposed single-step multiple testing procedures for EL that asymptotically control the family-wise error rate with Monte Carlo simulations or bootstrap. Wang and Yang (2018) applied the F-calibrated EL statistics to the Benjamini–Hochberg procedure (Benjamini and Hochberg 1995) to control the false discovery rate.

3. Overview of melt

The latest stable release of **melt** is available from the CRAN at `https://CRAN.R-project.org/package=melt`. The development version, hosted by the rOpenSci, is on GitHub at `https://github.com/ropensci/melt`. Computational tasks are implemented in parallel using **OpenMP** (Dagum and Menon 1998) API in C++ with the **Rcpp** (Eddelbuettel and Balamuta 2018) and **RcppEigen** (Bates and Eddelbuettel 2013) packages to interface with R. Depending on the platform, the package can be compiled from source with support for **OpenMP**. The overall design of **melt** adopts the functional object-oriented programming approach (Chambers 2014) with S4 classes and methods. Every function in the package is either a wrapper that creates a single instance of an object or a method that can be applied to a class object. The workflow of the package consists of three steps: (1) Fitting a model, (2) examining and diagnosing the fitted model, and (3) testing hypotheses with the model. Four functions are available to build a model object whose names start with the prefix `el_`, which stands for empirical likelihood. A summary of the functions is provided below.

- `el_mean()`: Creates an '`EL`' object for the mean.
- `el_sd()`: Creates a '`SD`' object for the standard deviation.
- `el_lm()`: Creates an '`LM`' object for the linear model.
- `el_glm()`: Creates a '`GLM`' object for the generalized linear model. It does not support grouped data.

For univariate data, `el_mean()` corresponds to `t.test()` in the **stats** package. The `el_lm()` and `el_glm()` functions correspond to `lm()` and `glm()`, respectively.

All model objects inherit from class '`EL`', and a description of the slots in '`EL`' is given in Table 1. Notably, the slot `optim` is a '`list`' with the following four components that summarize the optimization results:

Slot	Class	Description	Accessor
`optim`	'`list`'	Optimization results.	`getOptim()`
`logp`	'`numeric`'	Log probabilities of empirical likelihood.	`logProb()`
`logl`	'`numeric`'	Empirical log-likelihood.	`logL()`
`loglr`	'`numeric`'	Empirical log-likelihood ratio.	`logLR()`
`statistic`	'`numeric`'	Minus twice the empirical log-likelihood ratio.	`chisq()`
`df`	'`integer`'	Degrees of freedom associated with the `statistic`.	`getDF()`
`pval`	'`numeric`'	p value of the `statistic`.	`pVal()`
`nobs`	'`integer`'	Number of observations.	`nobs()`
`weights`	'`numeric`'	Re-scaled weights used for model fitting.	`weights()`
`coefficients`	'`numeric`'	MELE of the parameters.	`coef()`

Table 1: A description of some of the slots in an '`EL`' object. A full explanation of the class and slots can be found in the documentation of `EL-class` in the package.

- `par`: A numeric vector for the user-supplied parameter value θ where EL is evaluated.
- `lambda`: A numeric vector for the Lagrange multiplier λ.
- `iterations`: A single integer for the number of iterations performed.
- `convergence`: A single logical for the convergence status. It is either `TRUE` or `FALSE`.

Note that `par` is fixed in the evaluation of EL and should not be confused with the MELE, which is stored in the `coefficients` slot. The optimization is performed with respect to `lambda`, so `iterations` and `convergence` need to be understood in terms of `lambda`. Here we make a distinction between EL evaluation and EL optimization. The EL optimization refers to the constrained EL problem discussed in Section 2.3 and corresponds to another class '`CEL`' that directly extends '`EL`'. The `optim` slot in a '`CEL`' object has the same components. However, the optimization results are now interpreted in terms of `par`, the solution to the constrained problem. The '`LM`' and '`GLM`' classes contain '`CEL`', meaning that a constrained optimization is performed initially when `el_lm()` or `el_glm()` is called. In order to avoid confusion, the '`CEL`' class only distinguishes EL optimization from EL evaluation, and the user does not directly interact with a '`CEL`' object. Instead, the `optim` slot of every model object contains a single logical `cstr` that indicates whether EL optimization is performed or not. Once `par` is obtained through evaluation or optimization, it uniquely determines `lambda` and, in turn, `logl` and `loglr`. Then `statistic` is equivalent to `-2 * loglr` and has an asymptotic chi-square distribution under the null hypothesis, with the associated `df` and `pval`. All four model fitting functions above accept an optional argument `weights` for weighted data. A vector of weights is then re-scaled internally for numerical stability in the computation of weighted EL (Glenn and Zhao 2007). Although `weights()` and `coef()` can extract `weights` and `coefficients`, these slots are mainly stored for subsequent analyses and methods.

In the next step, the following methods can be applied to an object that inherits from '`EL`' to evaluate the model fit or compute summary statistics:

- `conv()`: Extracts convergence status from a model. The distinction between the EL evaluation and EL optimization applies here as well. It can be used to check the convex hull constraint indirectly.
- `confint()`: Computes confidence intervals for model parameters.
- `confreg()`: Computes a two-dimensional confidence region for model parameters. It returns an object of class '`ConfregEL`' where a subsequent `plot()` method is applicable.
- `eld()`: Computes empirical likelihood displacement in Equation 8 for model diagnostics and outlier detection. It returns an object of class '`ELD`' where a subsequent `plot()` method is applicable.

Lastly, we introduce the two main functions of **melt** that perform hypothesis testing. These generic methods take an '`EL`' object with other arguments that specify the problem in Equation 11.

- `elt()`: Tests a linear hypothesis with EL. It returns an object of class '`ELT`' that contains the test statistic, the critical value, and the level of the test. Several calibration options discussed in Section 2.2 are available, and the p value is computed by the calibration method chosen.
- `elmt()`: Tests multiple linear hypotheses simultaneously with EL. Each test can be considered as one instance of `elt()`, where the marginal test statistic has an asymptotic chi-square distribution under the corresponding null hypothesis. It returns an object of class '`ELMT`' with slots similar to those in '`ELT`'.

An '`ELT`' object also has the `optim` slot, which does not necessarily correspond to the EL optimization. The user can supply an arbitrary parameter value to test, reducing the problem to the EL evaluation. The `elmt()` function applies the single-step multiple testing procedure of Kim *et al.* (2023). The multiplicity-adjusted critical value and p values are estimated by Monte Carlo simulation. All model objects that inherit from '`EL`', '`ELT`', and '`ELMT`' support `print()` and `summary()` methods.

Note that every step of the workflow involves possibly multiple EL evaluations or optimizations. Hence, it is necessary to flexibly control the details of the execution and computation at hand. All model fitting functions and most methods accept an optional argument `control`, which allows the user to specify the control parameters. Only an object of class '`ControlEL`' can be supplied as `control` to ensure validity and avoid unexpected errors. Some of the slots in '`ControlEL`' are described in Table 2. This '`ControlEL`' object is stored in every model object, so any subsequent method can use those parameters unless the user overwrites them with new values. Another wrapper, `el_control()`, is available to construct a '`ControlEL`' object and specify the parameters. The default values are shown below.

```
el_control(maxit = 200L, maxit_l = 25L, tol = 1e-06, tol_l = 1e-06,
  step = NULL, th = NULL, verbose = FALSE, keep_data = TRUE, nthreads,
  seed = NULL, b = 10000L, m = 1000000L)
```

Specifically, `nthreads` specifies the number of threads for parallel computation via **OpenMP** (if available). By default, it is set to half the available threads and affects the following

Slot	Class	Description
`maxit`	'`integer`'	Maximum number of iterations for the EL optimization.
`maxit_l`	'`integer`'	Maximum number of iterations for the EL evaluation.
`tol`	'`numeric`'	Convergence tolerance for the EL optimization.
`tol_l`	'`numeric`'	Convergence tolerance for the EL evaluation.
`step`	'`numeric`'	Step size for projected gradient descent method in the EL optimization.
`th`	'`numeric`'	Threshold for the negative empirical log-likelihood ratio. The iteration stops if the value exceeds the threshold.
`nthreads`	`integer`	Number of threads for parallel computation.

Table 2: A description of some of the slots in an '`ControlEL`' object. A full explanation of the class and slots can be found in the documentation of `ControlEL-class` or `el_control()` in the package.

functions: `confint()`, `confreg()`, `el_lm()`, `el_glm()`, `eld()`, and `elt()`. For better performance, it is generally recommended in most platforms to limit the number of threads to the number of physical cores. The argument `seed` sets the seed for random number generation. It defaults to a random integer generated from 1 to the maximum integer supported by R on the machine, which is determined by `set.seed()`. For fast parallel random number generation and compatibility with **OpenMP**, the Xoshiro256+ pseudo-random number generator (period $2^{256} - 1$) of Blackman and Vigna (2021) is used internally with the **dqrng** package (Stubner 2023).

4. Usage

4.1. Model building

For a simple illustration of building a model, we apply `el_mean()` to the synthetic classification problem data `synth.tr` from the **MASS** package (Venables and Ripley 2002). The `synth.tr` object is a '`data.frame`' with 250 rows and three columns. We select two columns `xs` and `ys`, the x and y coordinates, to build an EL model with two-dimensional mean parameter. The resulting '`data.frame`' is denoted by `data`. The **dplyr** package (Wickham, François, Henry, and Müller 2023) and the **ggplot2** package (Wickham 2016) are used to aid data manipulation and visualization.

```
R> library("melt")
R> library("MASS")
R> library("dplyr")
R> library("ggplot2")
R> data("synth.tr", package = "MASS")
R> data <- dplyr::select(synth.tr, c(xs, ys))
```

With the focus on `xs` and `ys`, we first visualize the domain of the EL function with the convex hull constraint in Figure 1. Any parameter value inside the convex hull leads to proper EL

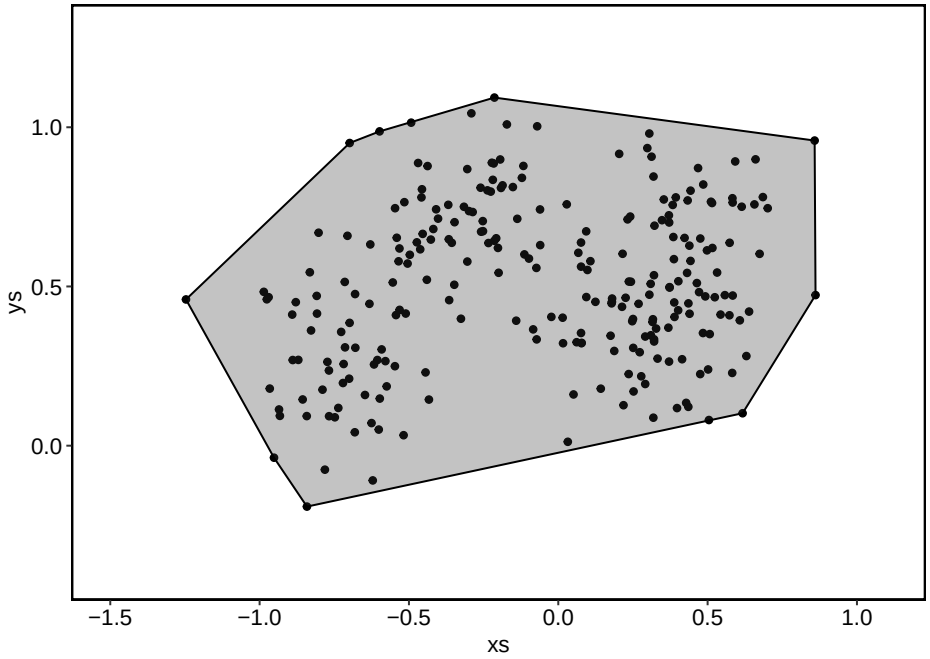

Figure 1: Scatter plot of `ys` versus `xs` in the `synth.tr`. The convex hull of the observations is shaded in gray.

evaluation. We specify `c(0, 0.5)` as `par` in `el_mean()` and build an '`EL`' object with the `data`.

```
R> fit_mean <- el_mean(data, par = c(0, 0.5))
```

The `data` object is implicitly coerced into a '`matrix`' since `el_mean()` takes a numeric '`matrix`' as an input for the data. Basic `print()` and `show()` methods display relevant information about an '`EL`' object.

```
R> fit_mean

  Empirical Likelihood

Model: mean

Maximum EL estimates:
     xs      ys
-0.0728  0.5044

Chisq: 6.16, df: 2, Pr(>Chisq): 0.046
EL evaluation: converged
```

The asymptotic chi-square statistic is displayed, along with the associated degrees of freedom and the p value. The `coef()` method extracts the MELE, which can be easily computed in this

case by using `colMeans(data)`. We note that the MELE is computed independently of the `par` specified by the user. Knowing the MELE makes it straightforward for the user to build a model with any valid `par` when the user is more interested in a subsequent application of `elt()` or `elmt()` to the fitted '`EL`' object. We can also specify an arbitrary `weight` in `el_mean()` for weighted EL evaluation. Then the MELE is the weighted average of the observations. The re-scaled weights returned by `weights()` add up to the total number of observations.

Next, we consider an infeasible parameter value `c(1, 0.5)` outside the convex hull to show how `el_control()` interacts with the model fitting functions through `control` argument. By employing the pseudo logarithm function in Equation 6, the evaluation algorithm continues until the iteration reaches `maxit_l` or the negative empirical log-likelihood ratio exceeds `th`. Setting a large `th` for the infeasible value, we observe that the algorithm hits the `maxit` with each element of `lambda` diverging quickly.

```
R> ctrl <- el_control(maxit_l = 50, th = 10000)
R> fit2_mean <- el_mean(data, par = c(1, 0.5), control = ctrl)
R> logL(fit2_mean)

[1] -10001

R> logLR(fit2_mean)

[1] -8621

R> getOptim(fit2_mean)

$par
 xs  ys
1.0 0.5

$lambda
[1] -9.909e+14  2.757e+14

$iterations
[1] 50

$convergence
[1] FALSE

$cstr
[1] FALSE
```

We generate a surface plot of the empirical log-likelihood ratio on the grid of Figure 1. The boundary of the convex hull separates the feasible region from the infeasible region in Figure 2.

A similar process applies to the other model fitting functions, except that `el_lm()` and `el_glm()` require a '`formula`' object for model specification. In addition, **melt** contains another function `el_eval()` to perform the EL evaluation for other general estimating functions. For example, consider the mean and standard deviation denoted by $\theta = (\mu, \sigma)$. For a given value of θ, we evaluate the estimating function $g(X_i, \theta) = (X_i - \mu, (X_i - \mu)^2 - \sigma^2)$ with the available data $X_1, \ldots, X_n$. The `el_eval()` function takes a '`matrix`' argument `g`, where each row corresponds to $g(X_i, \theta)$.

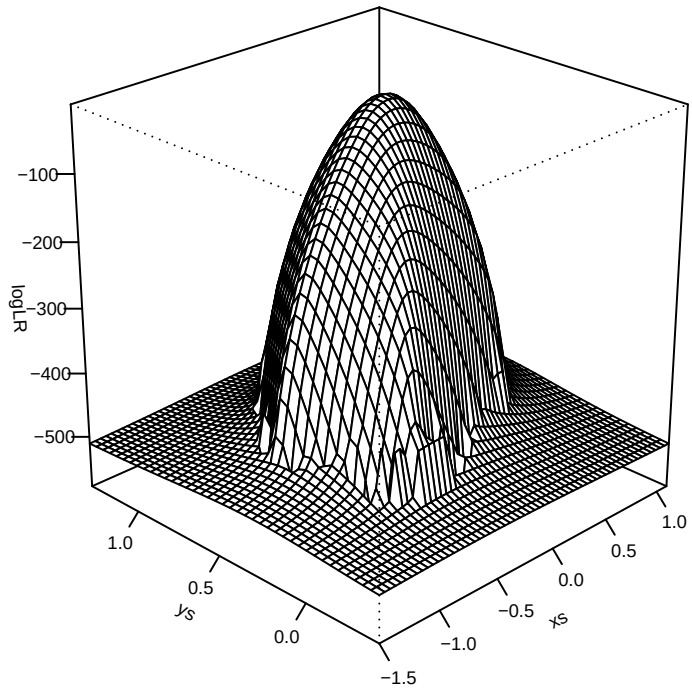

Figure 2: Surface plot of empirical log-likelihood ratio obtained from `synth.tr` with `el_mean()`. The argument `th` is set to 400.

```
R> mu <- 0
R> sigma <- 1
R> set.seed(123526)
R> x <- rnorm(100)
R> g <- matrix(c(x - mu, (x - mu)^2 - sigma^2), ncol = 2)
R> fit_eval <- el_eval(g)
R> fit_eval$pval

[1] 0.4646
```

Although the user can supply a custom `g`, `el_eval()` is not the main function of the package. The `el_eval()` function returns a '`list`' with the same components as in an '`EL`' object, but no other methods are applicable further. The scope is also limited to just-identified estimating functions. For more flexible and over-identified estimating functions, it is recommended to use other packages, e.g., **gmm** or **momentfit**.

4.2. Linear regression analysis

We illustrate the use of `el_lm()` for regression analysis with the crime rates data `UScrime` available in **MASS**. Here we update the control parameters for significance tests of the coefficients.

```
R> data("UScrime", package = "MASS")
R> ctrl <- el_control(maxit = 1000, nthreads = 2)
R> (fit_lm <- el_lm(y ~ Pop + Ineq, data = UScrime, control = ctrl))
```

```
  Empirical Likelihood

Model: lm

Maximum EL estimates:
(Intercept)         Pop        Ineq
    1046.75        3.25       -1.34

Chisq: 14, df: 2, Pr(>Chisq): 0.000933
Constrained EL: converged
```

The `print()` method also applies and shows the MELE, the overall model test result, and the convergence status. The estimates are obtained from `lm.fit()`. The hypothesis for the overall test is that all the parameters except the intercept are 0. The convergence status shows that a constrained optimization is performed in testing the hypothesis. The EL evaluation applies to the test and the convergence status if the model does not include an intercept. The `conv()` method can be used to extract the convergence status. It is designed to return a single logical, which can be helpful in a control flow where the convergence status decides the course of action. The large chi-square value above implies that the data do not support the hypothesis, regardless of the convergence. Note that failure to converge does not necessarily indicate unreliable test results. Most commonly, the algorithm fails to converge if the additional constraint imposed by a hypothesis is incompatible with the convex hull constraint. The control parameters affect the test results as well. The `summary()` method reports more details, such as the results of significance tests, where each test involves solving a constrained EL problem.

```
R> summary(fit_lm)

  Empirical Likelihood

Model: lm

Call:
el_lm(formula = y ~ Pop + Ineq, data = UScrime, control = ctrl)

Number of observations: 47
Number of parameters: 3

Parameter values under the null hypothesis:
(Intercept)         Pop        Ineq
       1047           0           0

Lagrange multipliers:
[1]  3.50e-03  1.42e-05 -2.62e-05

Maximum EL estimates:
(Intercept)         Pop        Ineq
    1046.75        3.25       -1.34
```

```
logL: -188 , logLR: -6.98
Chisq: 14, df: 2, Pr(>Chisq): 0.000933
Constrained EL: converged

Coefficients:
            Estimate  Chisq Pr(>Chisq)
(Intercept)  1046.75 447.64    < 2e-16 ***
Pop             3.25   4.93    0.02647 *
Ineq           -1.34  13.65    0.00022 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These tests are all asymptotically pivotal without explicit studentization. As a result, the output does not have standard errors.

By iteratively solving constrained EL problems for a grid of parameter values, confidence intervals for the parameters can be calculated with `confint()`. The chi-square calibration is the default, but the user can specify a critical value `cv` optionally. Below we calculate asymptotic 95% confidence intervals.

```
R> confint(fit_lm)

               lower     upper
(Intercept) 579.7584 1698.9193
Pop           0.3492    6.3530
Ineq         -1.9453   -0.6872
```

Without standard errors and `vcov()` methods, the `lower` and `upper` confidence limits do not necessarily correspond to 2.5 and 97.5 percentiles, respectively. Similarly, we obtain confidence regions for two parameters with `confreg()`. Starting from the MELE, it computes the boundary points of a confidence region in full circle. An optional argument `npoints` controls the number of boundary points. The return value is a '`ConfregEL`' object containing a matrix whose rows consist of the points, and the `plot()` method visualizes the confidence region (Figure 3).

```
R> cr <- confreg(fit_lm, parm = c("Pop", "Ineq"), npoints = 200)
R> plot(cr, cex = 1.5, cex.axis = 1.5, cex.lab = 1.5, lwd = 2, tck = -0.01)
```

Finally, we apply `eld()` to detect influential observations and outliers. Aside from the model object, `eld()` only accepts the control parameters. By the leave-one-out method of ELD, an '`ELD`' object inherits from the base type '`numeric`', with the length equal to the number of observations in the data. Figure 4 shows the ELD values from the `plot()` method.

```
R> eld <- eld(fit_lm)
R> summary(eld)
R> plot(eld, cex = 1.5, cex.axis = 1.5, cex.lab = 1.5, lwd = 2, pch = 19,
+    tck = -0.01)
```

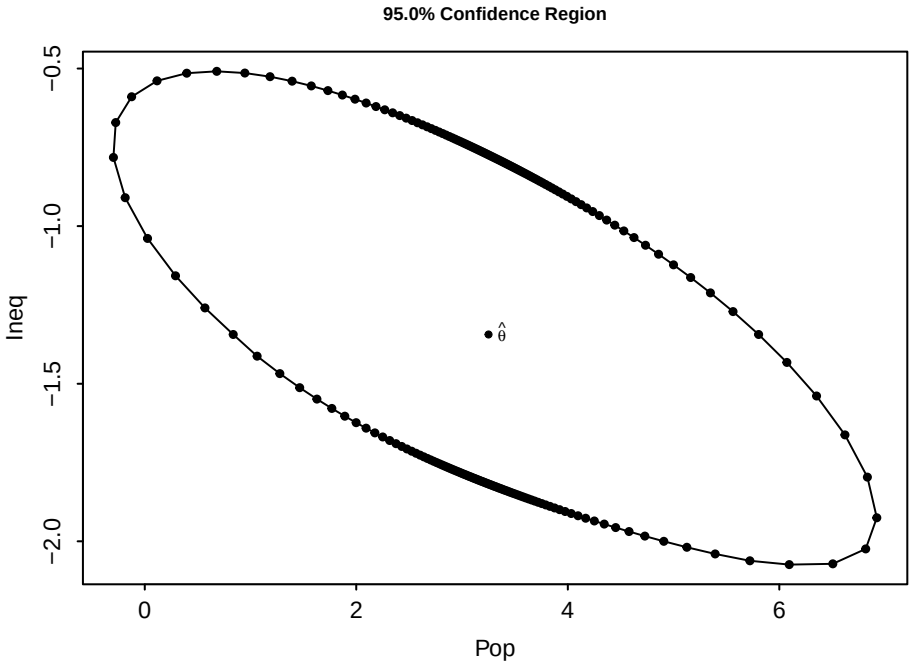

Figure 3: Scatter plot of the boundary points for asymptotic 95% confidence region of `Pop` and `Ineq` in `fit_lm`. At the center of the plot is the MELE $\hat{\theta}$.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.19    1.31    4.03    3.55   42.47
```

The code below shows that the observation with the largest ELD also has the largest Cook's distance from the same linear model fitted by `lm()`.

```
R> fit2_lm <- lm(y ~ Pop + Ineq, data = UScrime)
R> cd <- cooks.distance(fit2_lm)
R> all.equal(which.max(eld), which.max(cd), check.attributes = FALSE)

[1] TRUE
```

4.3. Hypothesis testing

Now we consider `elt()` for hypothesis testing, with the function prototype given below.

```
elt(object, rhs = NULL, lhs = NULL, alpha = 0.05, calibrate = "chisq",
  control = NULL)
```

The arguments `rhs` and `lhs` define a linear hypothesis and correspond to r and L in Equation 13, respectively. Therefore, either one or the other must be provided. The argument `lhs` takes a numeric matrix or a vector. Alternatively, a character vector can be supplied to symbolically specify a hypothesis, which is convenient when there are many variables. When

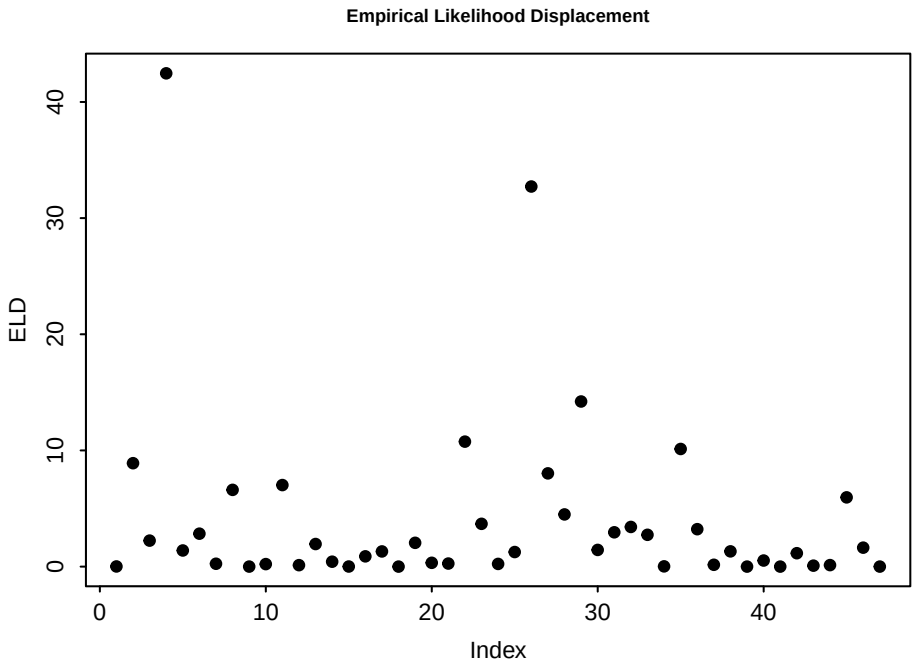

Figure 4: Scatter plot of empirical likelihood displacement versus observation index in `fit_lm`. The 4th observation has the largest value.

`lhs` is `NULL`, it performs the EL evaluation at $\theta = r$ by setting $L = I_p$, where I_p is the identity matrix of order p. When `rhs` is `NULL`, on the other hand, r is set to the zero vector automatically, and the EL optimization is performed with L. Technically, `elt()` can reproduce the test results from `fit_mean` in Section 4.1 and `fit_lm` in Section 4.2. Note the equivalence between the optimization results.

```
R> elt_mean <- elt(fit_mean, rhs = c(0, 0.5))
R> all.equal(getOptim(elt_mean), getOptim(fit_mean))

[1] TRUE

R> elt_lm <- elt(fit_lm, lhs = c("Pop", "Ineq"))
R> all.equal(getOptim(elt_lm), getOptim(fit_lm))

[1] TRUE
```

In addition to specifying an arbitrary linear hypothesis through `rhs` and `lhs`, extra arguments `alpha` and `calibrate` expand options for testing. The argument `alpha` controls the significance level determining the critical value, and `calibrate` chooses the calibration method. The `critVal()` method extracts the critical value from an 'ELT' object.

```
R> critVal(elt_mean)

[1] 5.991
```

We apply the F and bootstrap calibrations to `fit_mean` at a significance level of 0.05. The number of threads is increased to four with 100000 bootstrap replicates in `el_control()`.

```
R> ctrl <- el_control(maxit = 10000, tol = 1e-04, nthreads = 4, b = 100000,
+    step = 1e-05)
R> (elt_mean_f <- elt(fit_mean, rhs = c(0, 0.5), calibrate = "F",
+    control = ctrl))


  Empirical Likelihood Test

Hypothesis:
xs = 0.0
ys = 0.5

Significance level: 0.05, Calibration: F

Statistic: 6.16, Critical value: 6.09
p-value: 0.0484
EL evaluation: converged


R> (elt_mean_boot <- elt(fit_mean, rhs = c(0, 0.5), calibrate = "boot",
+    control = ctrl))


  Empirical Likelihood Test

Hypothesis:
xs = 0.0
ys = 0.5

Significance level: 0.05, Calibration: Bootstrap

Statistic: 6.16, Critical value: 6.06
p-value: 0.0476
EL evaluation: converged
```

The above output shows that the F and bootstrap calibrations tend to produce slightly larger critical values than the chi-square calibration. These values can be used as the `cv` argument in `confint()` and `confreg()`, improving coverage probabilities when the sample size is small.

Next, we compare `elt()` with `lht()` in **car** that computes an asymptotic chi-square statistic from Wald tests. The two functions have similar syntax with comparable outputs. For illustration, we fit a logistic regression model to the U.S. women's labor-force participation data `Mroz` from the **carData** package (Fox, Weisberg, and Price 2022) with `el_glm()` and `glm()`. We include all variables of `carData` in the model with the binary response variable `lfp`, which stands for labor-force participation. See the documentation of `carData` for a detailed description of the variables.

```
R> library("car")
R> data("Mroz", package = "carData")
R> fit_glm <- el_glm(lfp ~ ., family = binomial(link = "logit"),
+    data = Mroz, control = ctrl)
R> fit2_glm <- glm(lfp ~ ., family = binomial(link = "logit"), data = Mroz)
```

Asymptotic 95% confidence intervals from `confint()` can be compared with the ones from `confint.glm()` in the **MASS** package.

```
R> matrix(c(confint(fit_glm), confint(fit2_glm)), ncol = 4,
+    dimnames = list(
+      c(names(coef(fit2_glm))),
+      c("EL_lower", "EL_upper", "MASS_2.5%", "MASS_97.5%")
+    ))
```

```
            EL_lower EL_upper MASS_2.5% MASS_97.5%
(Intercept)  2.27606  4.09139    1.9370    4.46631
k5          -1.79757 -1.14646   -1.8609   -1.08747
k618        -0.18119  0.05263   -0.1984    0.06867
age         -0.07024 -0.05535   -0.0883   -0.03814
wcyes        0.41881  1.20724    0.3610    1.26378
hcyes       -0.23781  0.46629   -0.2920    0.51679
lwg          0.32517  0.91532    0.3140    0.90698
inc         -0.04985 -0.01970   -0.0510   -0.01877
```

We employ `coef()` to extract only the results of significance tests from the output of `summary()`.

```
R> coef(summary(fit_glm))
```

```
            Estimate   Chisq Pr(>Chisq)
(Intercept)  3.18214 539.769 2.116e-119
k5          -1.46291  85.631  2.169e-20
k618        -0.06457   1.174  2.785e-01
age         -0.06287 544.866 1.648e-120
wcyes        0.80727  16.705  4.366e-05
hcyes        0.11173   0.402  5.261e-01
lwg          0.60469  19.768  8.743e-06
inc         -0.03445  22.996  1.624e-06
```

Based on the estimates and p values above, we test two hypotheses that involve different classes of `lhs`: 1) `wc` = `hc` and 2) `k5` = -1.5 and `k618` = 0. Wald tests are performed by specifying `test = "Chisq"` in `lht()`.

```
R> lhs <- c(0, 0, 0, 0, 1, -1, 0, 0)
R> elt_glm <- elt(fit_glm, lhs = lhs)
R> lht_glm <- lht(fit2_glm, hypothesis.matrix = lhs, test = "Chisq")
R> lhs2 <- rbind(
```

```
+    c(0, 1, 0, 0, 0, 0, 0, 0),
+    c(0, 0, 1, 0, 0, 0, 0, 0))
R> rhs2 <- c(-1.5, 0)
R> elt2_glm <- elt(fit_glm, rhs = rhs2, lhs = lhs2)
R> lht2_glm <- lht(fit2_glm, hypothesis.matrix = lhs2, rhs = rhs2,
+    test = "Chisq")
```

For comparison, we extract the chi-square statistics and p values using `chisq()` and `pVal()`. The results are presented below.

```
R> matrix(c(chisq(elt_glm), pVal(elt_glm),
+      lht_glm$Chisq[2], lht_glm$`Pr(>Chisq)`[2]),
+    nrow = 2, byrow = TRUE,
+    dimnames = list(c("EL", "Wald"), c("Chisq", "Pr(>Chisq)")))

     Chisq Pr(>Chisq)
EL   3.634    0.05660
Wald 3.536    0.06004

R> matrix(c(chisq(elt2_glm), pVal(elt2_glm),
+      lht2_glm$Chisq[2], lht2_glm$`Pr(>Chisq)`[2]),
+    nrow = 2, byrow = TRUE,
+    dimnames = list(c("EL", "Wald"), c("Chisq", "Pr(>Chisq)")))

     Chisq Pr(>Chisq)
EL   1.144     0.5643
Wald 1.011     0.6032
```

The two tests provide similar results with a sample size of 753, which is not surprising given the asymptotic equivalence between these tests (see Qin and Lawless 1995, and references therein).

5. Case study

This section presents a more in-depth data analysis using EL with an internal dataset of **melt**, `thiamethoxam`, from Obregon, Pederson, Taylor, and Poveda (2022). Thiamethoxam is a widely used neonicotinoid pesticide that translocates through plants, leaving residues in crops. Since pesticides can also affect non-target organisms such as pollinators, it is important to maintain a balance between pest management and pollinator protection to maximize crop yield. Obregon *et al.* (2022) aimed to test how different application methods of thiamethoxam and plant variety impact pest control, bee visits, yield, and pesticide residues in flowers of squash crops. Squash crops rely on bee pollination to yield fruits (Knapp and Osborne 2019), and the striped cucumber beetle is the major pest for squash crops (Haber, Wallingford, Gretenberger, Ramirez Bonilla, Vinchesi-Vahl, and Weber 2021). Obregon *et al.* (2022) conducted a field experiment with two varieties that differ in their attractiveness to striped cucumber beetles: (1) Golden Zucchini (preferred by the beetle) and (2) Success PM straightneck summer squash (not preferred by the beetle). Also, the following four thiamethoxam application

methods were used: (1) In-furrow application after sowing, (2) foliar spray application three weeks after sowing, (3) seed treatment, and (4) no insecticides. Specifically, a quasi-Poisson regression model with a log link function was fit to examine the effects of plant variety and thiamethoxam application methods on the number of bee visits. The statistical significance of each variable was also tested, followed by Tukey's honest significant difference post hoc tests with the **agricolae** package (de Mendiburu 2023) for pairwise comparisons among the plant varieties and the application methods.

Following the original approach of Obregon *et al.* (2022), our goal is to conduct relevant tests with EL, focusing on performing multiple comparisons and constructing simultaneous confidence intervals. First, `thiamethoxam` is a '`data.frame`' with 165 observations and 11 variables. A summary of `thiamethoxam` is provided below.

```
R> data("thiamethoxam")
R> summary(thiamethoxam)
```

```
    trt       var      rep        fruit          avg_mass
 None  :41   SPM:82   1:24   Min.   : 1.00   Min.   :102
 Spray :40   GZ :83   2:24   1st Qu.: 4.00   1st Qu.:236
 Furrow:42            3:21   Median : 5.75   Median :310
 Seed  :42            4:24   Mean   : 6.11   Mean   :330
                      5:24   3rd Qu.: 7.25   3rd Qu.:401
                      6:24   Max.   :13.00   Max.   :724
                      7:24
     mass           yield           visit          foliage
 Min.   :  987   Min.   : 2778   Min.   : 0.75   Min.   :0.00014
 1st Qu.: 2751   1st Qu.: 7191   1st Qu.: 5.75   1st Qu.:0.00106
 Median : 5729   Median :11998   Median : 8.25   Median :0.00274
 Mean   : 6638   Mean   :13465   Mean   : 8.70   Mean   :0.00980
 3rd Qu.: 9673   3rd Qu.:17418   3rd Qu.:11.00   3rd Qu.:0.01199
 Max.   :16016   Max.   :34790   Max.   :23.50   Max.   :0.07131

     scb          defoliation
 Min.   : 0.125   Min.   : 0.00
 1st Qu.: 1.250   1st Qu.: 0.75
 Median : 1.938   Median : 2.13
 Mean   : 2.772   Mean   : 7.72
 3rd Qu.: 3.125   3rd Qu.:12.50
 Max.   :22.875   Max.   :48.75
 NA's   :3
```

The variables `trt` and `var` are '`factor`' variables for the application methods and the plant varieties, respectively. The `visit` variable denotes the number of bee visits per plot. The ridgeline plot in Figure 5 created by the **ggridges** package (Wilke 2022) shows distinct distributions of `visit` by `trt` and `var`. Note that the ranges of `visit` differ by `trt`. The seed treatment (`Seed`) records the largest number of visits among the methods compared to no treatment (`None`). As for the variety, Success PM (`SPM`) tends to have a larger number of visits than Golden Zucchini (`GZ`). Considering `visit` as our response variable, we also include `fruit`

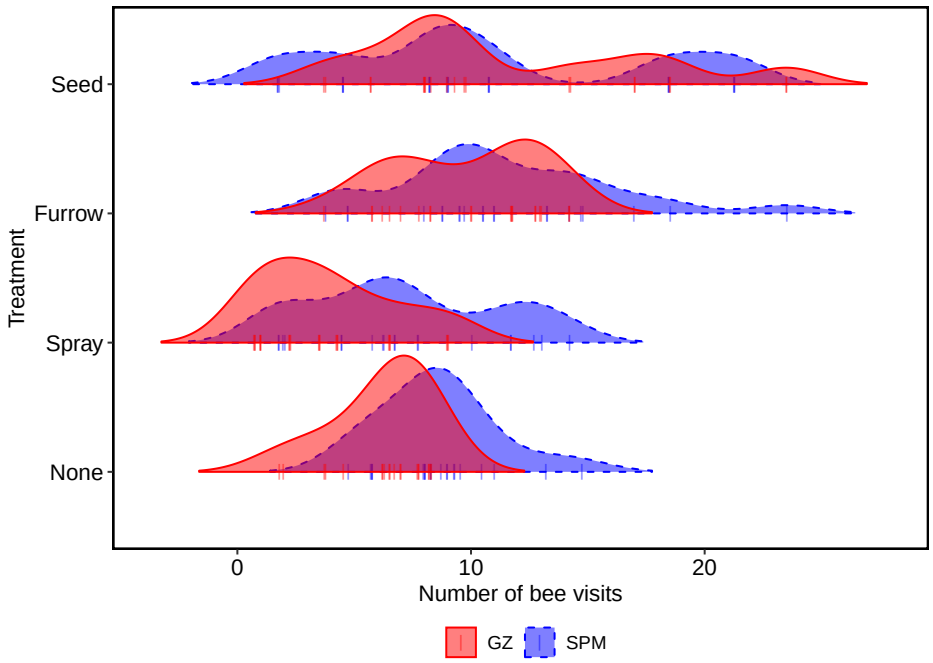

Figure 5: Ridgeline plot showing the densities of the number of bee visits (`visit`), grouped by the application methods (`trt`) and plant varieties(`var`). Solid red and dashed blue lines correspond to Golden Zucchini (`GZ`) and Success PM (`SPM`), respectively. Rugs show jittered data points.

(average number of fruits per plant) and `defoliation` (percentage defoliation) in our model as numeric variables. Particularly, Obregon *et al.* (2022) conducted a path analysis with the **piecewiseSEM** package (Lefcheck 2016), showing that the percentage defoliation significantly reduces the number of visits.

Next, we fit a quasi-Poisson regression model with a log link function using `el_glm()` to obtain a '`QGLM`' model object.

```
R> fit3_glm <- el_glm(visit ~ trt + var + fruit + defoliation,
+    family = quasipoisson(link = "log"), data = thiamethoxam,
+    control = ctrl)
R> print(summary(fit3_glm), width.cutoff = 50)

  Empirical Likelihood

Model: glm (quasipoisson family with log link)

Call:
el_glm(formula = visit ~ trt + var + fruit + defoliation,
    family = quasipoisson(link = "log"), data = thiamethoxam,
    control = ctrl)

Number of observations: 165
```

```
Number of parameters: 7

Parameter values under the null hypothesis:
(Intercept)    trtSpray   trtFurrow     trtSeed       varGZ
       1.97        0.00        0.00        0.00        0.00
      fruit defoliation         phi
       0.00        0.00        1.73

Lagrange multipliers:
[1] -0.2032 -0.1863  0.0183  0.1450 -0.1746  0.1096 -0.0487 -0.0877

Maximum EL estimates:
(Intercept)    trtSpray   trtFurrow     trtSeed       varGZ
     1.9723     -0.1128      0.0800      0.3179     -0.2109
      fruit defoliation
     0.0514     -0.0204

logL: -910 , logLR: -67.2
Chisq: 134, df: 6, Pr(>Chisq): <2e-16
Constrained EL: converged

Coefficients:
            Estimate  Chisq Pr(>Chisq)
(Intercept)   1.9723 421.87    < 2e-16 ***
trtSpray     -0.1128   1.68    0.19489
trtFurrow     0.0800   1.01    0.31404
trtSeed       0.3179  11.95    0.00055 ***
varGZ        -0.2109   9.50    0.00206 **
fruit         0.0514  14.47    0.00014 ***
defoliation  -0.0204  27.15    1.9e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Dispersion for quasipoisson family: 1.726
```

The dispersion estimate corresponds to $\hat{\phi}$ in Equation 10. This estimate is smaller than the one obtained from `summary()` when applied to a '`glm`' object because the denominator in Equation 10 is n instead of $n - p$. The solution to the constrained EL problem also includes `phi`, which is not part of the overall model constraint. Both `fruit` and `defoliation` are significant, although the estimates are smaller than other variables. With only the level `Seed` being significant in `trt`, we assess the significance of `trt` by testing whether the coefficients are all zero. The output of `summary()` reports a small p value with a different solution from the overall model test.

```
R> elt2_glm <- elt(fit3_glm, lhs = c("trtSpray", "trtFurrow", "trtSeed"))
R> summary(elt2_glm)

  Empirical Likelihood Test
```

```
Hypothesis:
trtSpray = 0
trtFurrow = 0
trtSeed = 0

Significance level: 0.05, Calibration: Chi-square

Parameter values under the null hypothesis:
(Intercept)    trtSpray   trtFurrow     trtSeed       varGZ
     1.9732      0.0000      0.0000      0.0000     -0.2102
      fruit defoliation         phi
     0.0596     -0.0254      1.7270

Lagrange multipliers:
[1] -0.09787 -0.15872  0.12336  0.25170  0.00985 -0.00207  0.00769
[8]  0.02068

logL: -850, logLR: -7.34
Statistic: 14.7, Critical value: 7.81
p-value: 0.00211
Constrained EL: converged
```

Finally, we extend the hypothesis testing framework of Section 4.3 to multiple testing with `elmt()`, which can be directly applied to the fitted model object. Its syntax is similar to `elt()`, where `rhs` and `lhs` now specify multiple hypotheses.

```
elmt(object, rhs = NULL, lhs = NULL, alpha = 0.05, control = NULL)
```

For general hypotheses involving separate matrices, `elmt()` accepts '`list`' objects for `rhs` and `lhs`. The corresponding elements of `rhs` and `lhs` together form a hypothesis, as in Equation 13. The `elmt()` function employs a multivariate chi-square calibration technique based on Monte Carlo simulations to determine the common critical value. Details of multiple testing procedures are given in Kim *et al.* (2023). Continuing on the previous test result, we perform comparisons with the control, which is our primary interest. We set the overall significance level at 0.05.

```
R> elmt_glm <- elmt(fit3_glm, lhs = list("trtSpray", "trtFurrow", "trtSeed"))
R> summary(elmt_glm)

  Empirical Likelihood Multiple Tests

Overall significance level: 0.05

Calibration: Multivariate chi-square

Hypotheses:
              Estimate Chisq Df  p.adj
```

```
trtSpray = 0    -0.113  1.68  1 0.4635
trtFurrow = 0    0.080  1.01  1 0.6625
trtSeed = 0      0.318 11.95  1 0.0016 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Common critical value: 5.66
```

Note the use of a '`list`' for `lhs` by `elmt()`. While a character vector `lhs` acts as a single hypothesis for `elt()`, elements of `lhs` in `elmt()` define distinct hypotheses for convenience. The `Df` column shows the marginal chi-square degrees of freedom for each hypothesis.

Further, we compare the result with the output of `glht()` in **multcomp**, which relies on (asymptotic) multivariate normal and t distributions for simultaneous tests.

```
R> library("multcomp")
R> fit4_glm <- glm(visit ~ trt + var + fruit + defoliation,
+    family = quasipoisson(link = "log"), data = thiamethoxam)
R> fit4_glm$call <- NULL
R> glht_glm <- glht(fit4_glm,
+    linfct = mcp(trt = c("Spray = 0", "Furrow = 0", "Seed = 0")))
R> summary(glht_glm)

   Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: User-defined Contrasts

Linear Hypotheses:
            Estimate Std. Error z value Pr(>|z|)
Spray == 0    -0.113      0.124   -0.91   0.6995
Furrow == 0    0.080      0.111    0.72   0.8224
Seed == 0      0.318      0.103    3.10   0.0057 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

For the hypothesis `Seed` vs. `None`, the adjusted p values are 0.00243 for `elmt()` and 0.00563 for `glht()`. Both procedures reject this hypothesis at the overall level of 0.05 and conclude that only the seed treatment is significantly different from the control. Since each hypothesis conforms to a linear combination of the parameters, `confint()` can be applied to produce asymptotic 95% simultaneous confidence intervals. For an object of class '`ELMT`', `confint()` uses the common critical value computed by `elmt()`. Below we give the intervals from the two procedures.

```
R> confint(elmt_glm)

                lower   upper
trtSpray = 0  -0.3689 0.08084
```

```
trtFurrow = 0 -0.1141 0.26524
trtSeed = 0    0.1041 0.51379

R> glht_sci <- confint(glht_glm)$confint
R> attributes(glht_sci)[c("calpha", "conf.level")] <- NULL
R> glht_sci

       Estimate      lwr    upr
Spray  -0.11281 -0.40521 0.1796
Furrow  0.08001 -0.18256 0.3426
Seed    0.31794  0.07583 0.5601
```

6. Conclusion

Empirical likelihood enables a likelihood-driven style of inference without the restrictive distributional assumptions of parametric models. Perhaps more importantly, while being nonparametric, empirical likelihood retains some desirable properties of parametric likelihood. In many ways, it is an attractive and natural approach to estimation and hypothesis testing, but its use has been limited due to computational difficulties compared to other methods. The R package **melt** aims to bridge the gap and provide a unified framework for data analysis with empirical likelihood methods. The package is developed to conduct statistical inference routinely made in R with empirical likelihood. Mainly, hypothesis testing is available for various models with smooth estimating functions. Examples in this paper demonstrate the functionality of **melt**. We provide more examples and details on the package website https://docs.ropensci.org/melt/. Future work will focus on expanding the scope to additional estimating functions and models. The package structure and its adoption of S4 classes and methods are designed for extensibility. Optimization algorithms tailored to specific models can also be added in the process.

Acknowledgments

We thank Pierre Chausse and Alex Stringer for their comments and suggestions on the package during the rOpenSci review process. This work was supported by the U.S. National Science Foundation under Grants No. SES-1921523 and DMS-2015552.

References

Adimari G, Guolo A (2010). "A Note on the Asymptotic Behaviour of Empirical Likelihood Statistics." *Statistical Methods & Applications*, **19**(4), 463–476. doi:10.1007/s10260-010-0137-9.

Barton WH (2022). ***emplik2**: Empirical Likelihood Ratio Test for Two Samples with Censored Data*. R package version 1.32, URL https://CRAN.R-project.org/package=emplik2.

Bates D, Eddelbuettel D (2013). "Fast and Elegant Numerical Linear Algebra Using the **RcppEigen** Package." *Journal of Statistical Software*, **52**(5), 1–24. doi:10.18637/jss.v052.i05.

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B*, **57**(1), 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x.

Blackman D, Vigna S (2021). "Scrambled Linear Pseudorandom Number Generators." *ACM Transactions on Mathematical Software*, **47**(4). ISSN 0098-3500. doi:10.1145/3460772.

Chambers JM (2014). "Object-Oriented Programming, Functional Programming and R." *Statistical Science*, **29**(2), 167–180. doi:10.1214/13-sts452.

Chaudhuri S, Mondal D, Yin T (2017). "Hamiltonian Monte Carlo Sampling in Bayesian Empirical Likelihood Computation." *Journal of the Royal Statistical Society B*, **79**(1), 293–320. doi:10.1111/rssb.12164.

Chaussé P (2010). "Computing Generalized Method of Moments and Generalized Empirical Likelihood with R." *Journal of Statistical Software*, **34**(11), 1–35. doi:10.18637/jss.v034.i11.

Chaussé P (2023). ***momentfit**: Methods of Moments*. R package version 0.5, URL https://CRAN.R-project.org/package=momentfit.

Chen SX, Cui H (2003). "An Extended Empirical Likelihood for Generalized Linear Models." *Statistica Sinica*, **13**(1), 69–81.

Chen SX, Peng L, Qin YL (2009). "Effects of Data Dimension on Empirical Likelihood." *Biometrika*, **96**(3), 711–722. doi:10.1093/biomet/asp037.

Chen SX, Van Keilegom I (2009). "A Review on Empirical Likelihood Methods for Regression." *Test*, **18**(3), 415–447. doi:10.1007/s11749-009-0159-5.

Cook RD (1986). "Assessment of Local Influence." *Journal of the Royal Statistical Society B*, **48**(2), 133–155. doi:10.1111/j.2517-6161.1986.tb01398.x.

Dagum L, Menon R (1998). "OpenMP: An Industry Standard API for Shared-Memory Programming." *IEEE Computational Science and Engineering*, **5**(1), 46–55. doi:10.1109/99.660313.

de Mendiburu F (2023). ***agricolae**: Statistical Procedures for Agricultural Research*. R package version 1.3-7, URL https://CRAN.R-project.org/package=agricolae.

DiCiccio T, Hall P, Romano J (1991). "Empirical Likelihood Is Bartlett-Correctable." *The Annals of Statistics*, **19**(2), 1053–1061. doi:10.1214/aos/1176348137.

DiCiccio TJ, Romano JP (1990). "Nonparametric Confidence Limits by Resampling Methods and Least Favorable Families." *International Statistical Review*, **58**(1), 59–76. doi:10.2307/1403474.

Dickhaus T, Royen T (2015). "A Survey on Multivariate Chi-Square Distributions and Their Applications in Testing Multiple Hypotheses." *Statistics*, **49**(2), 427–454. doi:10.1080/02331888.2014.993639.

Dickhaus T, Sirotko-Sibirskaya N (2019). "Simultaneous Statistical Inference in Dynamic Factor Models: Chi-Square Approximation and Model-Based Bootstrap." *Computational Statistics & Data Analysis*, **129**, 30–46. doi:10.1016/j.csda.2018.08.012.

Eddelbuettel D, Balamuta JJ (2018). "Extending R with C++: A Brief Introduction to **Rcpp**." *The American Statistician*, **72**(1), 28–36. doi:10.1080/00031305.2017.1375990.

Efron B (1981). "Nonparametric Standard Errors and Confidence Intervals." *Canadian Journal of Statistics*, **9**(2), 139–158. doi:10.2307/3314608.

Fox J, Weisberg S (2019). *An R Companion to Applied Regression*. 3rd edition. Sage, Thousand Oaks. URL https://socialsciences.mcmaster.ca/jfox/Books/Companion/.

Fox J, Weisberg S, Price B (2022). ***carData**: Companion to Applied Regression Data Sets*. R package version 3.0-5, URL https://CRAN.R-project.org/package=carData.

Glenn NL, Zhao Y (2007). "Weighted Empirical Likelihood Estimates and Their Robustness Properties." *Computational Statistics & Data Analysis*, **51**(10), 5130–5141. doi:10.1016/j.csda.2006.07.032.

Haber AI, Wallingford AK, Grettenberger IM, Ramirez Bonilla JP, Vinchesi-Vahl AC, Weber DC (2021). "Striped Cucumber Beetle and Western Striped Cucumber Beetle (Coleoptera: Chrysomelidae)." *Journal of Integrated Pest Management*, **12**(1), 1–10. doi:10.1093/jipm/pmaa026.

Hall P, Scala BL (1990). "Methodology and Algorithms of Empirical Likelihood." *International Statistical Review*, **58**(2), 109–127. doi:10.2307/1403462.

Hansen LP (1982). "Large Sample Properties of Generalized Method of Moments Estimators." *Econometrica*, **50**(4), 1029–1054. doi:10.2307/1912775.

Hansen LP, Heaton J, Yaron A (1996). "Finite-Sample Properties of Some Alternative GMM Estimators." *Journal of Business & Economic Statistics*, **14**(3), 262–280. doi:10.2307/1392442.

Hjort NL, McKeague IW, Van Keilegom I (2009). "Extending the Scope of Empirical Likelihood." *The Annals of Statistics*, **37**(3), 1079–1111. doi:10.1214/07-aos555.

Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." *Biometrical Journal*, **50**(3), 346–363. doi:10.1002/bimj.200810425.

Imbens GW (1997). "One-Step Estimators for Over-Identified Generalized Method of Moments Models." *The Review of Economic Studies*, **64**(3), 359–383. doi:10.2307/2971718.

Jacod J, Sørensen M (2018). "A Review of Asymptotic Theory of Estimating Functions." *Statistical Inference for Stochastic Processes*, **21**(2), 415–434. doi:10.1007/s11203-018-9178-8.

Kien DT, Chaudhuri S, Wei NH (2017). ***elhmc**: Sampling from a Empirical Likelihood Bayesian Posterior of Parameters Using Hamiltonian Monte Carlo*. R package version 1.1.0, URL https://CRAN.R-project.org/package=elhmc.

Kim E (2024). ***melt**: Multiple Empirical Likelihood Tests*. R package version 1.11.0, URL https://CRAN.R-project.org/package=melt.

Kim E, MacEachern SN, Peruggia M (2023). "Empirical Likelihood for the Analysis of Experimental Designs." *Journal of Nonparametric Statistics*, **35**(4), 709–732. doi:10.1080/10485252.2023.2206919.

Kitamura Y (1997). "Empirical Likelihood Methods with Weakly Dependent Processes." *The Annals of Statistics*, **25**(5), 2084–2102. doi:10.1214/aos/1069362388.

Kitamura Y, Stutzer M (1997). "An Information-Theoretic Alternative to Generalized Method of Moments Estimation." *Econometrica*, **65**(4), 861–874. doi:10.2307/2171942.

Kitamura Y, Tripathi G, Ahn H (2004). "Empirical Likelihood-Based Inference in Conditional Moment Restriction Models." *Econometrica*, **72**(6), 1667–1714. doi:10.1111/j.1468-0262.2004.00550.x.

Knapp JL, Osborne JL (2019). "Cucurbits as a Model System for Crop Pollination Management." *Journal of Pollination Ecology*, **25**, 88–102. doi:10.26786/1920-7603(2019)535.

Kolaczyk ED (1994). "Empirical Likelihood for Generalized Linear Models." *Statistica Sinica*, **4**(1), 199–218.

Lazar NA (2005). "Assessing the Effect of Individual Data Points on Inference From Empirical Likelihood." *Journal of Computational and Graphical Statistics*, **14**(3), 626–642. doi:10.1198/106186005x59568.

Lefcheck JS (2016). "**piecewiseSEM**: Piecewise Structural Equation Modeling in R for Ecology, Evolution, and Systematics." *Methods in Ecology and Evolution*, **7**(5), 573–579. doi:10.1111/2041-210x.12512.

Li G, Li R, Zhou M (2005). *Empirical Likelihood in Survival Analysis*, chapter Empirical Likelihood in Survival Analysis, pp. 337–349. World Scientific. doi:10.1142/9789812567765_0020.

Newey WK, Smith RJ (2004). "Higher Order Properties of GMM and Generalized Empirical Likelihood Estimators." *Econometrica*, **72**(1), 219–255. doi:10.1111/j.1468-0262.2004.00482.x.

Nordman DJ, Lahiri SN (2014). "A Review of Empirical Likelihood Methods for Time Series." *Journal of Statistical Planning and Inference*, **155**, 1–18. doi:10.1016/j.jspi.2013.10.001.

Obregon D, Pederson G, Taylor A, Poveda K (2022). "The Pest Control and Pollinator Protection Dilemma: The Case of Thiamethoxam Prophylactic Applications in Squash Crops." *PLOS One*, **17**(5), 1–18. doi:10.1371/journal.pone.0267984.

Owen A (1988). "Empirical Likelihood Ratio Confidence Intervals for a Single Functional." *Biometrika*, **75**(2), 237–249. doi:10.1093/biomet/75.2.237.

Owen A (1990). "Empirical Likelihood Ratio Confidence Regions." *The Annals of Statistics*, **18**(1), 90–120. doi:10.1214/aos/1176347494.

Owen A (1991). "Empirical Likelihood for Linear Models." *The Annals of Statistics*, **19**(4), 1725–1747. doi:10.1214/aos/1176348368.

Owen A (2001). *Empirical Likelihood*. Chapman & Hall/CRC, New York. doi:10.1201/9781420036152.

Qin J, Lawless J (1994). "Empirical Likelihood and General Estimating Equations." *The Annals of Statistics*, **22**(1), 300–325. doi:10.1214/aos/1176325370.

Qin J, Lawless J (1995). "Estimating Equations, Empirical Likelihood and Constraints on Parameters." *Canadian Journal of Statistics*, **23**(2), 145–159. doi:10.2307/3315441.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Shen B, Wang M (2023). ***ELCIC**: The Empirical Likelihood-Based Consistent Information Criterion*. R package version 0.2.1, URL https://CRAN.R-project.org/package=ELCIC.

Smith RJ (1997). "Alternative Semi-Parametric Likelihood Approaches to Generalised Method of Moments Estimation." *The Economic Journal*, **107**(441), 503–519. doi:10.1111/j.0013-0133.1997.174.x.

Stein C (1956). "Efficient Nonparametric Testing and Estimation." In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 187–195. doi:10.1525/9780520313880.

Stubner R (2023). ***dqrng**: Fast Pseudo Random Number Generators*. R package version 0.3.1, URL https://CRAN.R-project.org/package=dqrng.

Tang CY, Wu TT (2014). "Nested Coordinate Descent Algorithms for Empirical Likelihood." *Journal of Statistical Computation and Simulation*, **84**(9), 1917–1930. doi:10.1080/00949655.2013.770514.

Valeinis J, Cers E (2022). ***EL**: Two-Sample Empirical Likelihood*. R package version 1.2, URL https://CRAN.R-project.org/package=EL.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. ISBN 0-387-95457-0, URL https://www.stats.ox.ac.uk/pub/MASS4/.

Wang L, Yang D (2018). "F-Distribution Calibrated Empirical Likelihood Ratio Tests for Multiple Hypothesis Testing." *Journal of Nonparametric Statistics*, **30**(3), 662–679. doi:10.1080/10485252.2018.1461867.

Wedderburn RWM (1974). "Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method." *Biometrika*, **61**(3), 439–447. doi:10.1093/biomet/61.3.439.

Wickham H (2016). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag. ISBN 978-3-319-24277-4. doi:10.1007/978-0-387-98141-3.

Wickham H, François R, Henry L, Müller K (2023). ***dplyr**: A Grammar of Data Manipulation*. R package version 1.1.3, URL https://CRAN.R-project.org/package=dplyr.

Wilke CO (2022). ***ggridges**: Ridgeline Plots in **ggplot2***. R package version 0.5.4, URL https://CRAN.R-project.org/package=ggridges.

Yuan KH, Jennrich RI (1998). "Asymptotics of Estimating Equations under Natural Conditions." *Journal of Multivariate Analysis*, **65**(2), 245–260. doi:10.1006/jmva.1997.1731.

Zhou M (2015). *Empirical Likelihood Method in Survival Analysis*. Chapman & Hall/CRC. doi:10.1201/b18598.

Zhou M (2023). ***emplik**: Empirical Likelihood Ratio for Censored/Truncated Data*. R package version 1.3.1, URL https://CRAN.R-project.org/package=emplik.

Zhu H, Ibrahim JG, Tang N, Zhang H (2008). "Diagnostic Measures for Empirical Likelihood of General Estimating Equations." *Biometrika*, **95**(2), 489–507. doi:10.1093/biomet/asm094.

Affiliation:

Eunseop Kim, Steven N. MacEachern, Mario Peruggia
Department of Statistics
The Ohio State University
1958 Neil Ave.
Columbus, OH 43210, United States of America
E-mail: kim.7302@osu.edu, snm@stat.osu.edu, peruggia@stat.osu.edu

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
February 2024, Volume 108, Issue 5 *Submitted:* 2022-09-20
doi:10.18637/jss.v108.i05 *Accepted:* 2023-07-01